

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-222384
(43)Date of publication of application : 11.08.2000

(51)Int.Cl.

G06F 17/10
G06F 7/00
G06T 1/20

(21)Application number : 11-022934
(22)Date of filing : 29.01.1999

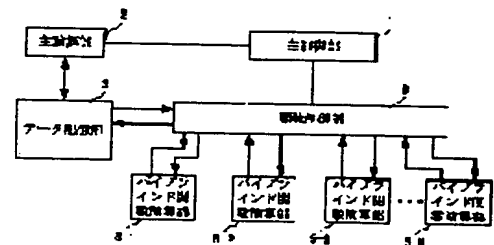
(71)Applicant : YAMATAKE CORP
(72)Inventor : MORIKAWA MAKOTO

(54) ARITHMETIC PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To provide an arithmetic processor quickly operating the maximum data processing using plural functions.

SOLUTION: This arithmetic processor is provided with pipe lined function arithmetic parts 5-1-5-N. A function controlling part 6 is provided between a main control part 1 and the pipe lined function arithmetic parts 5-1-5-N, and the pipe lined function arithmetic parts pine lined function arithmetic group) to be executed and the execution sequence are designated from the main control part 1 to the function controlling part 6. The function controlling part 6 reads data to be processed from a data storing part, and inputs the data to the pipe lined function arithmetic group (the pipe lined function arithmetic parts cascade- connected in the designated execution sequence), and the data of the arithmetic processed results from the pipe lined function arithmetic group are written in the data storing part 3.



LEGAL STATUS

[Date of request for examination] 28.09.2001
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-222384

(P2000-222384A)

(43) 公開日 平成12年8月11日 (2000.8.11)

(51) Int.Cl.	識別記号	F I	テマート (参考)
G 0 6 F	17/10	G 0 6 F 15/31	Z 5 B 0 2 2
	7/00	7/00	A 5 B 0 5 6
G 0 6 T	1/20	15/66	L 5 B 0 5 7

審査請求 未請求 請求項の数 3 O L (全 13 頁)

(21) 出願番号 特願平11-22934

(22) 出願日 平成11年1月29日 (1999.1.29)

(71) 出願人 000006666

株式会社山武

東京都渋谷区渋谷2丁目12番19号

(72) 発明者 森川 誠

東京都渋谷区渋谷2丁目12番19号 株式会
社山武内

(74) 代理人 100064621

弁理士 山川 政樹

Fターム (参考) 5B022 AA01 BA00 DA02 FA01

5B056 AA04 BB13 BB21 FF16 HH03

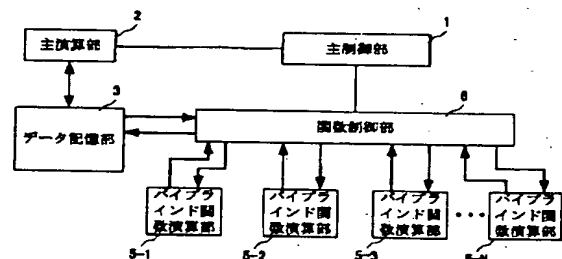
5B057 CC05 CH04 CH05 CH11

(54) 【発明の名称】 演算処理装置

(57) 【要約】

【課題】 多数の関数を用いる多大なデータ処理を高速で行う。

【解決手段】 バイブラインド関数演算部5-1~5-Nを設ける。主制御部1とバイブラインド関数演算部5-1~5-Nとの間に関数制御部6を設け、主制御部1より関数制御部6に対して実行すべきバイブラインド関数演算部 (バイブラインド関数演算群) およびその実行順序を指定する。関数制御部6は、データ記憶部3から処理すべきデータを読み出し、バイブラインド関数演算群 (指定の実行順序で縦続接続されたバイブラインド関数演算部) に投入し、このバイブラインド関数演算群からの演算処理結果のデータをデータ記憶部3に書き込む。



【特許請求の範囲】

【請求項1】 処理すべきデータが格納されたデータ記憶部と、

入力データの演算処理を終了する前に新たな入力データの演算処理を開始することの可能な複数のパイプライン関数演算部と、

これらパイプライン関数演算部の中から実行すべきパイプライン関数演算部をパイプライン関数演算群として選択のうえその実行順序を指定する主制御部と、
この主制御部によって指定された実行順序に従って前記パイプライン関数演算群のパイプライン関数演算部を継続接続させ、その処理データの受け渡しを制御する一方、前記データ記憶部から処理すべきデータを読み出して前記パイプライン関数演算群に入力すると共にこのパイプライン関数演算群からの演算処理結果のデータを前記データ記憶部に書き込む関数制御部とを備えたことを特徴とする演算処理装置。

【請求項2】 処理すべきデータが格納された第1のデータ記憶部と、

処理すべきデータが格納される第2のデータ記憶部と、
入力データの演算処理を終了する前に新たな入力データの演算処理を開始することの可能な複数のパイプライン関数演算部と、

前記第2のデータ記憶部から処理すべきデータを読み出し、この読み出したデータに対して前記パイプライン関数演算部で行う演算とは別個の主演算を行い、その演算処理結果のデータを前記第2のデータ記憶部に書き込む主演算部と、

前記パイプライン関数演算部の中から前記第1のデータ記憶部に格納されている処理すべきデータに対して実行すべきパイプライン関数演算部を第1のパイプライン関数演算群として選択のうえその実行順序を指定する一方、前記第2のデータ記憶部に格納されている前記主演算部での演算処理結果のデータに対して実行すべきパイプライン関数演算部を第2のパイプライン関数演算群として選択のうえその実行順序を指定する主制御部と、

この主制御部によって指定された実行順序に従って前記第1のパイプライン関数演算群のパイプライン関数演算部を継続接続させ、その処理データの受け渡しを制御する一方、前記第1のデータ記憶部から処理すべきデータを読み出して前記第1のパイプライン関数演算群に入力すると共にこの第1のパイプライン関数演算群からの演算処理結果のデータを前記主演算部で処理すべきデータとして前記第2のデータ記憶部に書き込む機能と、前記主制御部によって指定された実行順序に従って前記第2のパイプライン関数演算群のパイプライン関数演算部を継続接続させ、その処理データの受け渡しを制御する一方、前記演算部での演算処理結果のデータを前記第2のデータ記憶部から読み出して前記第2の

パイプライン関数演算群に入力すると共にこの第2のパイプライン関数演算群からの演算処理結果のデータを前記第1のデータ記憶部に書き込む機能とを有する関数制御部とを備えたことを特徴とする演算処理装置。

【請求項3】 処理すべきデータが格納された第1のデータ記憶部と、

処理すべきデータが格納される第2および第3のデータ記憶部と、

入力データの演算処理を終了する前に新たな入力データの演算処理を開始することの可能な複数のパイプライン関数演算部と、

前記第2および第3のデータ記憶部から処理すべきデータをデータ順に応じて互い違いに読み出し、この読み出したデータに対して前記パイプライン関数演算部で行う演算とは別個の主演算を行い、その演算処理結果のデータを読み出し元のデータ記憶部に書き込む主演算部と、

前記パイプライン関数演算部の中から前記第1のデータ記憶部に格納されている処理すべきデータに対して実行すべきパイプライン関数演算部を第1のパイプライン関数演算群として選択のうえその実行順序を指定する一方、前記第2および第3のデータ記憶部に格納されている前記主演算部での演算処理結果のデータに対して実行すべきパイプライン関数演算部を第2のパイプライン関数演算群として選択のうえその実行順序を指定する主制御部と、

この主制御部によって指定された実行順序に従って前記第1のパイプライン関数演算群のパイプライン関数演算部を継続接続させ、その処理データの受け渡しを制御する一方、前記第1のデータ記憶部から処理すべきデータを読み出して前記第1のパイプライン関数演算群に入力すると共にこの第1のパイプライン関数演算群からの演算処理結果のデータを前記主演算部で処理すべきデータとして前記第2および第3のデータ記憶部にデータ順に応じて互い違いに書き込む機能と、前記主制御部によって指定された実行順序に従って前記第2のパイプライン関数演算群のパイプライン関数演算部を継続接続させ、その処理データの受け渡しを制御する一方、前記主演算部での演算処理結果のデータを前記第2および第3のデータ記憶部からデータ順に応じて互い違いに読み出して前記第2のパイプライン関数演算群に入力すると共にこの第2のパイプライン関数演算群からの演算処理結果のデータを前記第1のデータ記憶部に書き込む機能とを有する関数制御部とを備えたことを特徴とする演算処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、画像処理などに用いて好適な演算処理装置に関するものである。

【0002】

【従来の技術】本出願人が開発を進めている画像処理装置では、画像データに対して拡大、縮小、回転、FFT（高速フーリエ変換）、DCT（離散コサイン変換）、相関計算などの主演算の他に、オフセット除去などの前処理や正規化処理などの後処理を行っている。例えば、位相限定方式のパターンマッチング装置等で、FFT処理を行う前に画像データにオフセット分を除去する前処理や、複素数を位相情報に変換する後処理を行っている。

【0003】図6にこの画像処理装置の要部構成を示す。同図において、1は主制御部、2はFFT処理を行う主演算部、3はデータ記憶部、4-1~4-Nは関数演算部である。データ記憶部3には処理すべき多数のデータが格納されている。関数演算部4-1~4-Nは現在演算中の入力データの演算処理を終了してからでないと新たな入力データの演算処理を開始することはできない。

【0004】この画像処理装置では次のようにしてFFT処理が行われる。主制御部1は、主演算部2でのFFT処理の前に、データ記憶部3に格納されている処理すべきデータに対して関数演算部4-1~4-Nを利用して前処理を行わせる。この場合、主制御部1は、関数演算部4-1~4-Nの中から実行すべき関数演算部を制御信号線S1~SNを介して選択する。ここでは、例えば、その実行順序を4-1→4-2→4-3として、関数演算部4-1、4-2、4-3を選択するものとする。

【0005】次に、主制御部1は、データ記憶部3から1番目のデータを読み出し、データバスDBを介して関数演算部4-1へ送る。関数演算部4-1での演算処理が終了すれば、その演算処理結果をデータバスDBを介してデータ記憶部3へ書き込んだうえ、データバスDBを介して関数演算部4-2へ送る。関数演算部4-2での演算処理が終了すれば、その演算処理結果をデータバスDBを介してデータ記憶部3へ書き込んだうえ、データバスDBを介して関数演算部4-3へ送る。関数演算部4-3での演算処理が終了すれば、その演算処理結果のデータを前処理完了データとしてデータ記憶部3に書き込む。そして、2番目のデータをデータ記憶部3から読み出し、1番目のデータと同様の演算シーケンスを施し、処理すべき全てのデータが完了するまでこの演算シーケンスを繰り返す。

【0006】なお、1つのデータの前処理が完了（データ記憶部3への書き込み）してから、次のデータの前処理を開始（関数演算部4-1へのデータの供与）する理由の1つとして、複数の関数演算部4が同時にアクティブになると、データバスDB上でデータ同士の衝突が発生し動作できなくなる虞れがあることが挙げられる。

【0007】K個のデータの前処理が完了すると、主演算部2は、主制御部1の指示に従い、データ記憶部3に

格納されている前処理完了データを読み出し、この前処理完了データに対してFFT処理を施し、このFFT処理を施したデータ（FFT完了データ）をデータ記憶部3に書き込む。全てのデータのFFT処理が完了すると、主制御部1は、データ記憶部3に格納されているFFT完了データに対し、関数演算部4-1~4-Nの中から実行すべき関数演算部およびその実行順序を指定のうえ、前処理と同様にして後処理を行わせる。

【0008】

【発明が解決しようとする課題】しかしながら、このような従来の画像処理装置では、前処理や後処理に必要な関数演算部が増える毎に、（個々の関数の処理時間）×処理データ数（通常は処理対象の画素数）の処理時間が加算されるため、処理時間が比例して増加してしまい、多数の関数を用いる画像処理を高速化することが困難であった。

【0009】例えば、K個のデータについて前処理を行う場合、同期回路での処理を想定すると、次のような処理の流れになる。なお、以下では、データ記憶部3からのリードおよびライト動作は通常同時には処理できず、実際にはそれぞれ1サイクルずつ加算される場合もあるが、説明の簡略化のため、リードおよびライトサイクルの時間は関数演算部での演算処理時間（サイクル数） S_n に含まれているものとする。

【0010】データ1：リード（0）→関数演算部4-1での演算処理（ S_1 ）→関数演算部4-2での演算処理（ S_1+S_2 ）→関数演算部4-3での演算処理（ $S_1+S_2+S_3$ ）→データ1：ライト（ $S_1+S_2+S_3$ ）→データ2：リード（ $S_1+S_2+S_3$ ）→関数演算部4-1での演算処理（ $2S_1+S_2+S_3$ ）→関数演算部4-2での演算処理（ $2S_1+2S_2+S_3$ ）→関数演算部4-3での演算処理（ $2S_1+2S_2+2S_3$ ）・・・データK：ライト（ $K \cdot (S_1+S_2+S_3)$ ）

【0011】すなわち、この例では、K個のデータについて前処理を行う場合、 $K \cdot (S_1+S_2+S_3)$ の処理時間を必要とする。ここで、n個の関数を実行する必要があるデータ数をKとすると、従来タイプの画像処理装置での前処理時間（あるいは後処理時間） T_0 は、下記（1）式で表される。

【0012】

【数1】

$$T_0 = K \cdot \sum_{n=1}^N S_n \quad \dots\dots (1)$$

【0013】このような式になるため、従来においては、前処理や後処理の関数で演算処理時間がかかるものや、処理関数の個数に比例して、全体の処理時間が増加してしまい、画像データのようにデータが多様で様々な演算処理を行う画像処理装置の高速化を実現することが

困難となっていた。

【0014】本発明はこのような課題を解決するためになされたもので、その目的とするところは、多数の関数を用いる多大なデータ処理を高速で行うことの可能な演算処理装置を提供することにある。

【0015】

【課題を解決するための手段】このような目的を達成するために、第1発明（請求項1に係る発明）は、データ記憶部と、複数のパイプラインド関数演算部と、主制御部と、関数制御部とを設け、パイプラインド関数演算部の中から実行すべきパイプラインド関数演算部をパイプラインド関数演算群として選択のうえその実行順序を指定するものとし、この指定された実行順序に従ってパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、データ記憶部から処理すべきデータを読み出してパイプラインド関数演算群に入力すると共にこのパイプラインド関数演算群からの演算処理結果のデータをデータ記憶部に書き込むようにしたものである。

【0016】この発明によれば、データ記憶部から処理すべきデータが読み出され、パイプラインド関数演算群に入力され、このパイプラインド関数演算群からの演算処理結果のデータがデータ記憶部に書き込まれる。ここで、パイプラインド関数演算群は、そのパイプラインド関数演算群を構成するパイプラインド関数部およびその実行順序が主制御部により指定され、この指定された実行順序に従ってパイプラインド関数演算群のパイプラインド関数演算部が縦続接続される。この縦続接続されたパイプラインド関数演算部の各々は、現在演算中の入力データの演算処理を終了する前に新たな入力データの演算処理を開始することが可能であり、パイプライン関数演算部での処理時間 S_n より速かに小さい投入待ち時間 X ($X \ll S_n$) をもってパイプラインド関数演算群へ次々にデータを入力することができる。

【0017】第2発明（請求項2に係る発明）は、第1のデータ記憶部と、第2のデータ記憶部と、複数のパイプラインド関数演算部と、主演算部と、主制御部と、関数制御部とを設け、パイプラインド関数演算部の中から第1のデータ記憶部に格納されている処理すべきデータに対して実行すべきパイプラインド関数演算部を第1のパイプラインド関数演算群として選択のうえその実行順序を指定するものとし、この指定された実行順序に従って第1のパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、第1のデータ記憶部から処理すべきデータを読み出して第1のパイプラインド関数演算群に入力すると共にこの第1のパイプラインド関数演算群からの演算処理結果のデータを主演算部で処理すべきデータとして第2のデータ記憶部に書き込むようにし、この第2のデータ記憶部から処理すべきデータを読み出し、この読み出したデータに対して主演算を行い、その演算処理結果のデータを第2のデータ記憶部に書き

込むようにし、また、第2のデータ記憶部に格納されている主演算部での演算処理結果のデータに対して実行すべきパイプラインド関数演算部を第2のパイプラインド関数演算群として選択のうえその実行順序を指定するものとし、この指定された実行順序に従って第2のパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、主演算部での演算処理結果のデータを第2のデータ記憶部から読み出して第2のパイプラインド関数演算群に入力すると共にこの第2のパイプラインド関数演算群からの演算処理結果のデータを第1のデータ記憶部に書き込むようにしたものである。

【0018】この発明によれば、第1のデータ記憶部から処理すべきデータが読み出され、第1のパイプラインド関数演算群に入力され、この第1のパイプラインド関数演算群からの演算処理結果のデータが第2のデータ記憶部に書き込まれる。そして、この第2のデータ記憶部に書き込まれたデータが読み出されて主演算が行われ、その演算処理結果のデータが第2のデータ記憶部に書き込まれる。この第2のデータ記憶部に書き込まれた主演算部での演算処理結果のデータは、第2のパイプラインド関数演算群に入力され、この第2のパイプラインド関数演算群からの演算処理結果のデータが第1のデータ記憶部に書き込まれる。

【0019】ここで、第1および第2のパイプラインド関数演算群は、そのパイプラインド関数演算群を構成するパイプラインド関数部およびその実行順序が主制御部により指定され、この指定された実行順序に従ってパイプラインド関数演算群のパイプラインド関数演算部が縦続接続される。この縦続接続されたパイプラインド関数演算部の各々は、現在演算中の入力データの演算処理を終了する前に新たな入力データの演算処理を開始することが可能であり、パイプライン関数演算部での処理時間 S_n より速かに小さい投入待ち時間 X ($X \ll S_n$) をもって第1および第2のパイプラインド関数演算群へ次々にデータを入力することができる。また、第2のデータ記憶部では主演算部との間でデータの読み出しと書き込みとが短時間で切り替わる可能性があるが、第1のデータ記憶部ではデータの読み出しと書き込みとが短時間で切り替わらない。

【0020】第3発明（請求項3に係る発明）は、第1のデータ記憶部と、第2のデータ記憶部と、第3のデータ記憶部と、複数のパイプラインド関数演算部と、主演算部と、主制御部と、関数制御部とを設け、パイプラインド関数演算部の中から第1のデータ記憶部に格納されている処理データに対して実行すべきパイプラインド関数演算部を第1のパイプラインド関数演算群として選択のうえその実行順序を指定するものとし、この指定された実行順序に従って第1のパイプラインド関数演算群のパイプラインド関数演算部を縦続接続させ、第1のデータ記憶部から処理すべきデータを読み出して第1のバイ

パイプライン関数演算群に入力すると共にこの第1のパイプライン関数演算群からの演算処理結果のデータを主演算部で処理すべきデータとして第2および第3のデータ記憶部にデータ順に応じて互い違いに書き込むようにし、この第2および第3のデータ記憶部から処理すべきデータをデータ順に応じて互い違いに読み出し、この読み出したデータに対して主演算を行い、その演算処理結果のデータを読み出し元のデータ記憶部に書き込むようにし、また、第2および第3のデータ記憶部に格納されている主演算部での演算処理結果のデータに対して実行すべきパイプライン関数演算部を第2のパイプライン関数演算群として選択のうえその実行順序を指定するものとし、この指定された実行順序に従って第2のパイプライン関数演算群のパイプライン関数演算部を縦続接続させ、主演算部での演算処理結果のデータを第2および第3のデータ記憶部からデータ順に応じて互い違いに読み出して第2のパイプライン関数演算群からの演算処理結果のデータを第1のデータ記憶部に書き込むようにしたものである。

【0021】この発明によれば、第1のデータ記憶部から処理すべきデータが読み出され、第1のパイプライン関数演算群に入力され、この第1のパイプライン関数演算群からの演算処理結果のデータがデータ順に応じて互い違いに第2および第3のデータ記憶部に書き込まれる。また、第1および第2のデータ記憶部に書き込まれたデータがデータ順に応じて互い違いに読み出されて主演算部による演算が行われ、この主演算部での演算処理結果のデータが読み出し元のデータ記憶部に書き込まれる。第1および第2のデータ記憶部に書き込まれた主演算部での演算処理結果のデータは、データ順に応じて互い違いに第2のパイプライン関数演算群に入力され、この第2のパイプライン関数演算群からの演算処理結果のデータが第1のデータ記憶部に書き込まれる。

【0022】ここで、第1および第2のパイプライン関数演算群は、そのパイプライン関数演算群を構成するパイプライン関数部およびその実行順序が主制御部により指定され、この指定された実行順序に従って第1および第2のパイプライン関数演算群のパイプライン関数演算部が縦続接続される。この縦続接続されたパイプライン関数演算部の各々は、現在演算中の入力データの演算処理を終了する前に新たな入力データの演算処理を開始することが可能であり、パイプライン関数演算部での処理時間 S_n より速かに小さい投入待ち時間 X ($X < S_n$) でもって第1および第2のパイプライン関数演算群へ次々にデータを入力することができる。

【0023】また、第2および第3のデータ記憶部では主演算部との間でデータの読み出しと書き込みとが短時間で切り替わる可能性があるが、第1のデータ記憶部ではデータの読み出しと書き込みとが短時間で切り替わら

ない。また、主演算部と第3のデータ記憶部（第2のデータ記憶部）との間で主演算処理を行っている間に、第2のデータ記憶部（第3のデータ記憶部）からの第2のパイプライン関数演算群を介する第1のデータ記憶部への演算処理結果のデータの書き込み、第1のデータ記憶部からの第1のパイプライン関数演算群を介する第2のデータ記憶部（第3のデータ記憶部）への演算処理結果のデータの書き込みを行うことができる。

【0024】

【発明の実施の形態】以下、本発明を実施の形態に基づき詳細に説明する。

【実施の形態1：第1発明】図1はこの発明の一実施の形態の要部を示すブロック図である。同図において、図6と同一符号は同一或いは同等構成要素を示し、その説明は省略する。

【0025】この実施の形態1では、従来の関数演算部4-1~4-Nに代えて、1入力1出力のパイプライン関数演算部5-1~5-Nを使用している。また、主制御部1とパイプライン関数演算部5-1~5-Nとの間に関数制御部6を設け、主制御部1より関数制御部6に対して実行すべきパイプライン関数演算部（パイプライン関数演算群）およびその実行順序を指定するようにしている。

【0026】パイプライン関数演算部5（5-1~5-N）は、現在演算中の入力データの演算処理を終了する前に新たな入力データの演算処理を開始することが可能な関数演算部（例えば、四則演算などの関数処理用）であり、1番目の処理対象データを入力後、その演算処理結果がまだ出力されていなくても（1データの関数演算に必要な時間 S_n が経過していなくても）、すぐ次のサイクルに2番目の処理対象データの入力が可能である。また、このパイプライン関数演算部5には、ある関数で演算が完了後、次の関数への受け渡しが可能ないようにハンドシェイク信号を出力するようなインターフェイスを構成しておく。

【0027】関数制御部6は、基本的にはマルチプレクサで構成されており、主制御部1によって選択されたパイプライン関数演算群のパイプライン関数演算部を指定された実行順序に従って縦続接続させ、その処理データの受け渡しを制御する一方、データ記憶部3から処理すべきデータを読み出してそのパイプライン関数演算群に入力すると共に、このパイプライン関数演算群からの演算処理結果のデータをデータ記憶部3に書き込む機能を有している。

【0028】この画像処理装置では次のようにしてFFT処理が行われる。主制御部1は、主演算部2でのFFT処理の前に、データ記憶部3に格納されている処理すべきデータに対してパイプライン関数演算部5-1~5-Nを利用して前処理を行わせる。

【0029】この場合、主制御部1は、関数制御部6に

対して、パイプライン関数演算部5-1~5-Nの中から前処理に際して実行すべきパイプライン関数演算部を第1のパイプライン関数演算群として選択のうえ、その実行順序を指定する。ここでは、例えば、パイプライン関数演算部5-1、5-2、5-3を第1のパイプライン関数演算群として選択し、その実行順序を5-1→5-2→5-3とするものとする。

【0030】関数制御部6は、この主制御部1によって指定された実行順序に従って第1のパイプライン関数演算群のパイプライン関数演算部5-1、5-2、5-3を縦続接続させ、データ記憶部3から処理すべき1番目のデータを読み出して第1のパイプライン関数演算群に入力する。この第1のパイプライン関数演算群に入力されたデータは、パイプライン関数演算部5-1→5-2→5-3の順に演算処理され、最後のパイプライン関数演算部5-3からの演算処理結果のデータが前処理完了データとしてデータ記憶部3に書き込まれる。

【0031】ここで、関数制御部6は、データ記憶部3から処理すべき1番目のデータを読み出した後、次のサイクルで2番目のデータを読み出して第1のパイプライン関数演算群に投入する。すなわち、データ記憶部3から1番目のデータを読み出して第1のパイプライン関数演算群に投入した後、パイプライン関数演算部での処理時間 S_n より遥かに短い投入待ち時間 X ($X \ll S_n$)でもって、データ記憶部3から2番目のデータを読み出して第1のパイプライン関数演算群に投入する。

【0032】第1のパイプライン関数演算群に投入された2番目のデータは、1番目のデータと同様にしてパイプライン関数演算部5-1→5-2→5-3の順に演算処理され、最後のパイプライン関数演算部5-3からの演算処理結果のデータが処理完了データとしてデータ記憶部3に書き込まれる。以下、同様の演算シーケンスを施し、処理すべき全てのデータが完了するまでこ

の演算シーケンスを繰り返す。

【0033】全てのデータの前処理が完了すると、主演算部2は、主制御部1の指示に従い、データ記憶部3に格納されている前処理完了データを読み出し、この前処理完了データに対してFFT処理を施し、このFFT処理を施したデータ（FFT完了データ）をデータ記憶部3に書き込む。

【0034】全てのデータのFFT処理が完了すると、主制御部1は、関数制御部6に対して、パイプライン関数演算部5-1~5-Nの中から後処理に際して実行すべきパイプライン関数演算部を第2のパイプライン関数演算群として選択のうえ、その実行順序を指定する。

【0035】関数制御部6は、この主制御部1によって選択された第2のパイプライン関数演算群のパイプライン関数演算部を指定された実行順序に従って縦続接続させ、前処理と同様にして、データ記憶部3に格納されているFFT完了データを第2のパイプライン関数演算群へ次々に投入して後処理を行わせ、その後処理完了データをデータ記憶部3に書き込む。

【0036】この実施の形態1では、指定された実行順序に従ってパイプライン関数演算群のパイプライン関数演算部を縦続接続させ、この縦続接続させたパイプライン関数演算部にデータを通して行くので、すなわち複数の関数演算が連続的にパイプライン処理されるので、バスの衝突等が発生することがない。このため、パイプライン関数演算群におけるパイプライン関数演算部の個数を n 個とし、それぞれのパイプライン関数演算部での処理時間を S_n 、処理すべきデータ数を K 個とすると、全てのデータをパイプライン関数演算群によって演算処理する時間（前処理時間/後処理時間） T_n は、下記（2）式で表される。

【0037】

【数2】

$$T_n = (K-1) \cdot X + \sum_{n=1}^n S_n \quad \dots\dots (2)$$

【0038】この（2）式と前述した従来タイプの（1）式とを比較して分かるように、この実施の形態1では、関数の必要数や画素数が増加しても、それに殆ど影響されずに、極めて高速に前処理や後処理を行うことが可能となる。

【0039】〔実例〕例えば、 512×512 画素の画像データがあり、これをFFT処理（主演算：処理時間20ms）する場合を考えてみる。FFT処理前に、予めオフセット分を除去し（前処理関数1：減算、処理時間40ns）、それを増幅するものとする（前処理関数2：乗算、処理時間80ns）。また、FFT処理結果の最大値を用いて正規化し（後処理関数1：除算、処理時間380ns）、ある値以下を0にする（後処理関数

2：しきい値処理、処理時間20ns）ものとする。これらの個々の処理関数の処理時間は、従来タイプの場合も本実施の形態の場合（新タイプ）も、同じ時間とする。また、主演算処理も同処理時間とする。

【0040】ここで、従来タイプも新タイプも、共に動作周波数50MHz（1サイクル=20ns）と仮定すると、それぞれの全処理時間 T_A および T_B は次のようになる。

【0041】〔従来タイプ： T_A 〕

前処理： $512 \times 512 \times (40+80) \text{ ns} = 31,5 \text{ ms}$

主演算：20ms

後処理： $512 \times 512 \times (380+20) \text{ ns} = 10$

4. 9ms

$TA = 31.5ms + 20ms + 104.9ms = 156.4ms$

6. 4ms

【0042】〔新タイプ:TB〕

前処理: $(512 \times 512 - 1) \times 20ns + (40 + 80)ns \approx 5.24ms$

主演算: 20ms

後処理: $(512 \times 512 - 1) \times 20ns + (380 + 20)ns \approx 5.24ms$

$TB = 5.24ms + 20ms + 5.24ms = 30.5ms$

【0043】この場合、従来タイプの全処理時間TAと新タイプの全処理時間TBとの差は $TA - TB = 125.9ms$ であり、新タイプの方が従来タイプよりも5倍以上高速に処理できることが確認できる。

【0044】ここで、特筆すべき点は前処理および後処理に要する時間で、新タイプでは従来タイプに比べ、6~20倍以上高速に処理可能となっている。この実例では、前処理よりも後処理の方が処理時間にして3.3倍大きいので、従来タイプでは後処理の方が3.3倍処理時間を必要としている。一方、新タイプでは、縦続接続可能なインターフェイスを持つパイプライン関数演算部を用いた処理構成のため、殆ど処理時間の増加にはつながらない。これは処理対象データ（ここでは、 512×512 ）が多いほど、その影響度は少なくなる。

【0045】関数単体の処理時間が増加しても影響が少ないのと同様に、処理関数の増加に対しても新タイプでは影響が少ない。そのため、新タイプでは、複雑な多数の関数処理が必要となるような前処理や後処理でも、処理時間の増加を殆ど伴わずに実行可能である。

【0046】〔実施の形態2〕実施の形態1では、データ記憶部3において、データの読み出しと書き込みとが交互に頻繁に切り替えられる。このため、データ記憶部3としては、データの読み出しと書き込みとの切り替えに際して殆ど待ち時間の生じない高性能の大容量メモリを使用する必要があり、高価となる。データの読み出しと書き込みとの切り替えに際して比較的待ち時間が生じる安価な大容量メモリをデータ記憶部3として使用すると、処理速度の劣化につながり、高速処理が阻害される。

【0047】そこで、この実施の形態2では、データ記憶部3として高性能の大容量メモリを使用しなくてもよいようにして（データ記憶部3として比較的待ち時間が生じる安価な大容量メモリの使用を可能として）、コストパフォーマンスの向上を図る。

【0048】図2はこの実施の形態2の要部を示すブロック図である。この実施の形態2では、データ記憶部3を第1のデータ記憶部とし、この第1のデータ記憶部3とは別に第2のデータ記憶部7を設けている。第1のデ

ータ記憶部3としては、データの読み出しと書き込みとの切り替えに際して比較的待ち時間が生じる安価な大容量メモリを使用する。第2のデータ記憶部7としては、データの読み出しと書き込みとの切り替えに際して殆ど待ち時間の生じない高性能の小容量メモリを使用する。また、主演算部2は大容量の第1のデータ記憶部3に対してではなく、小容量の第2のデータ記憶部7に対してアクセス可能に設ける。

【0049】この画像処理装置では次のようにしてFFT処理が行われる。主制御部1は、主演算部2でのFFT処理の前に、第1のデータ記憶部3に格納されている処理すべきデータに対してパイプライン関数演算部5-1~5-Nを利用して前処理を行わせる。

【0050】この場合、主制御部1は、関数制御部6に対して、パイプライン関数演算部5-1~5-Nの中から前処理に際して実行すべきパイプライン関数演算部を第1のパイプライン関数演算群として選択のうえ、その実行順序を指定する。

【0051】関数制御部6は、主制御部1によって指定された実行順序に従って第1のパイプライン関数演算群のパイプライン関数演算部を縦続接続させ、第1のデータ記憶部3から処理すべき1番目のデータを読み出して第1のパイプライン関数演算群に入力する。この第1のパイプライン関数演算群に入力されたデータは、縦続接続されたパイプライン関数演算部により順次演算処理され、この第1のパイプライン関数演算群からの演算処理結果のデータが前処理完了データとして第2のデータ記憶部7に書き込まれる。

【0052】ここで、関数制御部6は、第1のデータ記憶部3から処理すべき1番目のデータを読み出した後、次のサイクルで2番目のデータを読み出して第1のパイプライン関数演算群に投入する。第1のパイプライン関数演算群に投入された2番目のデータは、1番目のデータと同様にして、縦続接続されたパイプライン関数演算部により順次演算処理され、この第1のパイプライン関数演算群からの演算処理結果のデータが前処理完了データとして第2のデータ記憶部7に書き込まれる。

【0053】第2のデータ記憶部7に2つの前処理完了データが溜まると、主演算部2は、主制御部1の指示に従い、第2のデータ記憶部7に格納されている2つの前処理完了データを読み出し、この前処理完了データに対してFFT処理を施し、このFFT処理を施したデータ（FFT完了データ）を第2のデータ記憶部7に書き込む。

【0054】第2のデータ記憶部7における前処理完了データのFFT処理が終了すると、主制御部1は、関数制御部6に対して、パイプライン関数演算部5-1~5-Nの中から後処理に際して実行すべきパイプライン関数演算部を第2のパイプライン関数演算群として

選択のうえ、その実行順序を指定する。

【0055】関数制御部6は、主制御部1によって指定された実行順序に従って第2のバイブライント関数演算群のバイブライント関数演算部を縦続接続させ、前処理と同様にして、第2のデータ記憶部7に格納されているFFT完了データを第2のバイブライント関数演算群へ次々に投入し、後処理を行わせた後、第1のデータ記憶部3に書き込む。

【0056】そして、関数制御部6は、第1のデータ記憶部3に全ての後処理完了データを書き込んだ後、主制御部1によって指定される実行順序に従って第1のバイブライント関数演算群のバイブライント関数演算部を縦続接続させ、第1のデータ記憶部3に格納されている処理すべきデータの第1のバイブライント関数演算群への投入を再開する。

【0057】以下、同様にして、第1のバイブライント関数演算群を用いての前処理、主演算部2でのFFT処理、第2のバイブライント関数演算群を用いての後処理を繰り返すことによって、第1のデータ記憶部3に格納されている全ての処理すべきデータに対して前処理→FFT処理→後処理を施す。

【0058】この実施の形態2では、第1のデータ記憶部3において、データの読み出しと書き込みとを短時間で切り替える必要がなく、第1のデータ記憶部3としてデータの読み出しと書き込みとの切り替えに際して比較的時間が生じる安価な大容量メモリを使用することができる。

【0059】また、この実施の形態2では、第2のデータ記憶部7は主演算部2でのFFT処理のためにデータの読み出しと書き込みとが短時間に切り替わる場合でも殆ど待ち時間の生じない高性能のメモリを必要とするが、第2のデータ記憶部7に格納された前処理完了データは主演算部2によってFFT処理が施された後、FFT完了データとして第2のデータ記憶部7に格納されるものの、すぐに関数制御部6によって読み出され第2のバイブライント関数演算群に投入されるので、第2のデータ記憶部7のメモリ容量は小容量でよい。

【0060】これにより、第2のデータ記憶部7の追加によるコストアップ分が第1のデータ記憶部3のコストダウン分に吸収され、コストパフォーマンスが向上する。また、この実施の形態2では、第1のデータ記憶部3に他の装置からアクセスがある場合（例えば、画像入力部からの画像データ入力や画像出力部への画像データの出力など）でも、主演算部2での待ち時間が発生せず、処理速度が劣化しない。

【0061】〔実施の形態3〕実施の形態2では、第2のデータ記憶部7に前処理完了データを書き込んでいる間や第2のデータ記憶部7からFFT完了データを読み出している間は、主演算部2でのFFT処理を実行することができず、全演算処理に要する時間が長くなる。そ

こで、この実施の形態2では、データの転送時間の無駄を削減して、全演算処理に要する時間を短縮する。

【0062】図3はこの実施の形態3の要部を示すブロック図である。この実施の形態3では、データ記憶部3を第1のデータ記憶部とし、この第1のデータ記憶部3とは別に第2のデータ記憶部7と第3のデータ記憶部8を設けている。第1のデータ記憶部3としては、データの読み出しと書き込みとの短時間の切り替えに際して比較的時間が生じる安価な大容量メモリを使用する。第2のデータ記憶部7および第3のデータ記憶部8としては、データの読み出しと書き込みとの短時間の切り替えに際して殆ど待ち時間の生じない高性能の小容量メモリを使用する。また、主演算部2は大容量の第1のデータ記憶部3に対してではなく、小容量の第2のデータ記憶部7と第3のデータ記憶部8に対してアクセス可能に設ける。

【0063】この画像処理装置では次のようにしてFFT処理が行われる。主制御部1は、主演算部2でのFFT処理の前に、データ記憶部3に格納されている処理すべきデータに対してバイブライント関数演算部5-1〜5-Nを利用して前処理を行わせる。

【0064】この場合、主制御部1は、関数制御部6に対して、バイブライント関数演算部5-1〜5-Nの中から前処理に際して実行すべきバイブライント関数演算部を第1のバイブライント関数演算群として選択のうえ、その実行順序を指定する。

【0065】関数制御部6は、主制御部1によって指定された実行順序に従って第1のバイブライント関数演算群のバイブライント関数演算部を縦続接続させ、第1のデータ記憶部3から処理すべき1番目のデータを読み出して第1のバイブライント関数演算群に入力する。この第1のバイブライント関数演算群に入力されたデータは、縦続接続されたバイブライント関数演算部により順次演算処理され、この第1のバイブライント関数演算群からの演算処理結果のデータが前処理完了データ（1番目の前処理完了データ）として第2のデータ記憶部7に書き込まれる（図4（a）参照）。

【0066】ここで、関数制御部6は、第1のデータ記憶部3から処理すべき1番目の処理データを読み出した後、次のサイクルで2番目の処理すべきデータを読み出して第1のバイブライント関数演算群に投入する。第1のバイブライント関数演算群に投入された2番目のデータは、1番目のデータと同様にして、縦続接続されたバイブライント関数演算部により順次演算処理され、この第1のバイブライント関数演算群からの演算処理結果のデータが前処理完了データ（2番目の前処理完了データ）として第3のデータ記憶部8に書き込まれる（図4（b）参照）。

【0067】一方、主演算部2は、主制御部1からの指示に従い、第2のデータ記憶部7に格納されている1番

目の前処理完了データを読み出し、この1番目の前処理完了データに対してFFT処理を施す。すなわち、この場合、1番目の前処理完了データに対してFFT処理が行われている間に、空いている転送経路を利用して、2番目の前処理完了データが先読みされて第3のデータ記憶部8に書き込まれることになる。

【0068】次に、主演算部2は、主制御部1からの指示に従い、第3のデータ記憶部8に格納されている2番目の前処理完了データを読み出し、この2番目の前処理完了データに対してFFT処理を施す。この2番目の前処理完了データに対してFFT処理が行われている間に、関数制御部6は、空いている転送経路を利用して、第2のデータ記憶部7に格納されている1番目のFFT完了データを第2のバイパラインド関数演算群に投入し、後処理を行わせて、第1のデータ記憶部3に1番目の後処理完了データとして書き込む。そして、この後、関数制御部6は、第1のデータ記憶部3に格納されている3番目の処理すべきデータを第1のバイパラインド関数演算群に投入し、前処理を行わせて、第2のデータ記憶部7に3番目の前処理完了データとして書き込む(図4(c)参照)。

【0069】次に、主演算部2は、主制御部1からの指示に従い、第2のデータ記憶部7に格納されている3番目の前処理完了データを読み出し、この3番目の前処理完了データに対してFFT処理を施す。この3番目の前処理完了データに対してFFT処理が行われている間に、関数制御部6は、空いている転送経路を利用して、第3のデータ記憶部8に格納されている2番目のFFT完了データを第2のバイパラインド関数演算群に投入し、後処理を行わせて、第1のデータ記憶部3に2番目の後処理完了データとして書き込む。そして、この後、関数制御部6は、第1のデータ記憶部3に格納されている4番目の処理すべきデータを第1のバイパラインド関数演算群に投入し、前処理を行わせて、第3のデータ記憶部8に4番目の前処理完了データとして書き込む(図4(d)参照)。

【0070】以下、同様にして、図4(c)、図4(d)の処理を交互に行い、第1のバイパラインド関数演算群を用いての前処理、主演算部2でのFFT処理、第2のバイパラインド関数演算群を用いての後処理を繰り返すことによって、第1のデータ記憶部3に格納されている全ての処理すべきデータに対して前処理→FFT処理→後処理を施す。

【0071】この実施の形態3では、主演算部2と第3のデータ記憶部8との間でFFT処理を行っている間に、第2のデータ記憶部7からの第2のバイパラインド演算関数群を介する第1のデータ記憶部3への後処理完了データの書き込み、第1のデータ記憶部3からの第1のバイパラインド演算関数群を介する第2のデータ記憶部7への前処理完了データの書き込みを行うことができ

る。

【0072】また、主演算部2と第2のデータ記憶部7との間でFFT処理を行っている間に、第3のデータ記憶部8からの第2のバイパラインド演算関数群を介する第1のデータ記憶部3への後処理完了データの書き込み、第1のデータ記憶部3からの第1のバイパラインド演算関数群を介する第3のデータ記憶部8への前処理完了データの書き込みを行うことができる。

【0073】これにより、データの転送時間の無駄が削減され、主演算部2でのFFT処理が休みなく行われるものとなり、全演算処理にかかる時間が短縮されるようになる。

【0074】また、この実施の形態3では、第1のデータ記憶部3において、データの読み出しと書き込みとの切り替えが頻繁に繰り返されるように思われるが、「前処理のデータ転送時間+後処理のデータ転送時間」<主演算時間とすれば、第1のデータ記憶部3でのデータの読み出しと書き込みとを短時間で切り替える必要はなく、第1のデータ記憶部3としてデータの読み出しと書き込みとの短時間の切り替えに際して比較的待ち時間が生じる安価な大容量メモリを使用することができる。

【0075】また、この実施の形態3では、第2のデータ記憶部7(第3のデータ記憶部8)は主演算部2でのFFT処理のためにデータの読み出しと書き込みとを短時間で切り替える場合でも殆ど待ち時間の生じない高性能のメモリを必要とするが、第2のデータ記憶部7および第3のデータ記憶部8に格納された前処理完了データは主演算部2によってFFT処理が施された後、FFT完了データとして第2のデータ記憶部7および第3のデータ記憶部8に格納されるものの、すぐに関数制御部6によって読み出され第2のバイパラインド関数演算群に投入されるので、第2のデータ記憶部7および第3のデータ記憶部8のメモリ容量は小容量でよい。

【0076】なお、この実施の形態3では、説明を簡単とするために、第1のデータ記憶部3から1つつづつデータが読み出され、前処理された後、互い違いに第2のデータ記憶部7および第3のデータ記憶部8へ書き込まれるものとしたが、実際にはその投入時間を1サイクルずらした2つのデータがベアとして前処理された後、互い違いに第2のデータ記憶部7および第3のデータ記憶部8へ書き込まれる。主演算部2はこの2つの前処理完了データを読み出してFFT処理を施す。また、その投入時間を1サイクルずらした2つのFFT完了データがベアとして第2のデータ記憶部7および第3のデータ記憶部8から互い違いに読み出され、後処理された後、第1のデータ記憶部3に書き込まれる。

【0077】〔実施の形態2と実施の形態3の処理時間の比較〕実施の形態2(図2)において、第1のデータ記憶部3から第2のデータ記憶部7への転送をTR、第2のデータ記憶部7から第1のデータ記憶部3への転送

をTW、主演算部2でのFFT処理をFFTとすると、その処理状況は図5(a)に示すようになる。

【0078】実施の形態3(図3)において、第1のデータ記憶部3から第2のデータ記憶部7および第3のデータ記憶部8への転送をTR、第2のデータ記憶部7および第3のデータ記憶部8から第1のデータ記憶部3への転送をTW、主演算部2でのFFT処理をFFTとすると、その処理状況は図5(b)に示すようになる。

【0079】図5(a)でも図5(b)でもFFT、TR、TWの時間幅はそれぞれ同じとする(但し、 $TR+TW < FFT$)。すると、 $FFT1 \sim FFT128$ (2次元FFTの半分の時間(横方向か縦方向のみ))の時間は、図5(a)の場合には $128(TR+FFT+TW)$ 、図5(b)の場合には $TR+128FFT+TW$ となる。

【0080】この場合、図5(a)と図5(b)との差は $127TR+127TW$ となり、おおよそ $127TR+127TW$ だけ図5(b)、すなわち実施の形態3の方が高速に処理できることになる。

【0081】パイプライン関数の処理時間はTR、TWにほとんど含まれてしまう(実際は各パイプライン関数の段数の和だけTR、TWが増加する)。例えば、転送処理に256サイクルかかって、パイプライン関数に乗算(3段パイプライン)、加算(2段パイプライン)が転送と同時に縦続接続処理されているとすると、 $TR' = TR+3+2=261$ サイクルという具合になる。

【0082】なお、実施の形態1~3では、説明を簡単とするためにパイプライン関数演算部5-1~5-Nは1入力1出力としたが、一部の関数は2入力1出力、3入力1出力などとしてすることができる(例えば、2画像の差分を算出するような関数)。この場合、第1のパイプライン関数演算群や第2のパイプライン関数演算群において、その先頭のパイプライン関数演算部のみを複数入力1出力とする。複数入力1出力とした場合、他の入力を持つ必要があるので、投入待ち時間Xは、入力数をmとした場合、 $MAX(Xm)$ となる。

【0083】また、実施の形態1~3では、主演算部2においてFFT処理を行うものとしたが、拡大、縮小、回転、DCT、相関計算など各種の主演算に置き換えることが可能である。また、主演算は、パイプライン処理可能な演算であってもよい。しかし、完全にパイプライン処理できるのであれば、それは主演算ではなくパイプライン関数演算群で実現可能である。FFT処理の場合は、全体で考えると1入力1出力で処理することができず、途中で複数経路に分かれるなど複雑な処理となる。主演算部2としては、FFT処理のように、1入力1出力で処理することができない複雑な演算が適している。

【0084】また、実施の形態1~3は画像処理装置への適用例として説明したが、本発明は画像処理に限られ

るものではなく、各種の演算処理に適用可能である。また、実施の形態2および3では、処理すべきデータに対して前処理と後処理の両方を施す場合を示したが、どちらか一方だけの場合でもよい。

【0085】

【発明の効果】以上説明したことから明らかなように本発明によれば、第1発明では、データ記憶部から処理すべきデータが読み出され、パイプライン関数演算群に入力され、このパイプライン関数演算群からの演算処理結果のデータがデータ記憶部に書き込まれるものとなり、ここで、パイプライン関数演算群は、そのパイプライン関数演算群を構成するパイプライン関数部およびその実行順序が主制御部により指定され、この指定された実行順序に従ってパイプライン関数演算群のパイプライン関数演算部が縦続接続されるので、パイプライン関数演算部での処理時間 S_n より遥かに小さい投入待ち時間X($X < S_n$)でもってパイプライン関数演算群へ次々にデータを入力することができ、多数の関数を用いる多大なデータ処理を高速で行うことができるようになる。

【0086】第2発明では、第1のデータ記憶部から処理すべきデータが読み出され、第1のパイプライン関数演算群に入力され、この第1のパイプライン関数演算群からの演算処理結果のデータが第2のデータ記憶部に書き込まれ、この第2のデータ記憶部に書き込まれたデータが読み出されて主演算が行われ、その演算処理結果のデータが第2のデータ記憶部に書き込まれ、この第2のデータ記憶部に書き込まれた主演算部での演算処理結果のデータが第2のパイプライン関数演算群に入力され、この第2のパイプライン関数演算群からの演算処理結果のデータが第1のデータ記憶部に書き込まれるものとなり、ここで、第1および第2のパイプライン関数演算群は、そのパイプライン関数演算群を構成するパイプライン関数部およびその実行順序が主制御部により指定され、この指定された実行順序に従ってパイプライン関数演算群のパイプライン関数演算部が縦続接続されるので、パイプライン関数演算部での処理時間 S_n より遥かに小さい投入待ち時間X($X < S_n$)でもって第1および第2のパイプライン関数演算群へ次々にデータを入力することができ、多数の関数を用いる多大なデータ処理を高速で行うことができるようになる。

【0087】また、第2発明では、第2のデータ記憶部では主演算部との間でデータの読み出しと書き込みとが短時間で切り替わる可能性があるが、第1のデータ記憶部ではデータの読み出しと書き込みとが短時間で切り替わらず、第1のデータ記憶部としてデータの読み出しと書き込みとが短時間で切り替わる場合に比較的時間が生じる安価な大容量メモリを使用することができる。また、第2のデータ記憶部としてはデータの読み出しと

書き込みの短時間の切り替わりが生じて待ち時間が殆ど生じない高性能のメモリを必要とするが、第2のデータ記憶部に格納された主演算部で処理すべきデータは主演算部によって主演算が施された後、処理演算結果のデータとして第2のデータ記憶部に格納されるものの、すぐに関数制御部によって読み出され第2のバイブラインド関数演算群に投入されるので、第2のデータ記憶部のメモリ容量は小容量でよい。これにより、第2のデータ記憶部の追加によるコストアップ分が第1のデータ記憶部のコストダウン分に吸収され、コストパフォーマンスが向上する。

【0088】第3発明では、第1のデータ記憶部から処理すべきデータが読み出され、第1のバイブラインド関数演算群に入力され、この第1のバイブラインド関数演算群からの演算処理結果のデータがデータ順に応じて互い違いに第2および第3のデータ記憶部に書き込まれ、また、第1および第2のデータ記憶部に書き込まれたデータがデータ順に応じて互い違いに読み出されて主演算部による演算が行われ、この主演算部での演算処理結果のデータが読み出し元のデータ記憶部に書き込まれ、第1および第2のデータ記憶部に書き込まれた主演算部での演算処理結果のデータが、データ順に応じて互い違いに第2のバイブラインド関数演算群に入力され、この第2のバイブラインド関数演算群からの演算処理結果のデータが第1のデータ記憶部に書き込まれるものとなり、ここで、第1および第2のバイブラインド関数演算群は、そのバイブラインド関数演算群を構成するバイブラインド関数部およびその実行順序が主制御部により指定され、この指定された実行順序に従って第1および第2のバイブラインド関数演算群のバイブラインド関数演算部が継続接続されるので、バイブラインド関数演算部での処理時間 S_n より遥かに小さい投入待ち時間 X ($X \ll S_n$) でもって第1および第2のバイブラインド関数演算群へ次々にデータを入力することができ、多数の関数を用いる多大なデータ処理を高速で行うことができるようになる。

【0089】また、第3発明では、第2および第3のデータ記憶部では主演算部との間でデータの読み出しと書き込みとが短時間で切り替わる可能性があるが、第1のデータ記憶部ではデータの読み出しと書き込みとが短時間で切り替わらず、第1のデータ記憶部としてデータの読み出しと書き込みとの短時間の切り替えに際して比較的待ち時間が生じる安価な大容量メモリを使用すること

ができる。また、第2のデータ記憶部（第3のデータ記憶部）としてはデータの読み出しと書き込みの短時間の切り替わりが生じて待ち時間が殆ど生じない高性能のメモリを必要とするが、第2のデータ記憶部（第3のデータ記憶部）に格納された主演算部で処理すべきデータは主演算部によって主演算が施された後、処理演算結果のデータとして第2のデータ記憶部（第3のデータ記憶部）に格納されるものの、すぐに関数制御部によって読み出され第2のバイブラインド関数演算群に投入されるので、第2のデータ記憶部（第3のデータ記憶部）のメモリ容量は小容量でよい。これにより、第2および第3のデータ記憶部の追加によるコストアップ分が第1のデータ記憶部のコストダウン分に吸収され、コストパフォーマンスが向上する。

【0090】また、第3発明では、主演算部と第3のデータ記憶部（第2のデータ記憶部）との間で主演算処理を行っている間に、第2のデータ記憶部（第3のデータ記憶部）からの第2のバイブラインド関数演算群を介する第1のデータ記憶部への演算処理結果のデータの書き込み、第1のデータ記憶部からの第1のバイブラインド関数演算群を介する第2のデータ記憶部（第3のデータ記憶部）への演算処理結果のデータの書き込みを行うことができ、データの転送時間の無駄を削減して、全演算処理に要する時間を短縮することができる。

【図面の簡単な説明】

【図1】 本発明の一実施の形態（実施の形態1）の要部を示すブロック図である。

【図2】 本発明の他の実施の形態（実施の形態2）の要部を示すブロック図である。

【図3】 本発明の他の実施の形態（実施の形態3）の要部を示すブロック図である。

【図4】 実施の形態3における処理動作を説明する図である。

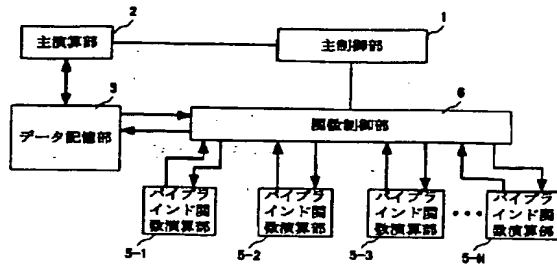
【図5】 実施の形態2と実施の形態3の処理時間の比較を説明する図である。

【図6】 従来の画像処理装置の要部を示すブロック図である。

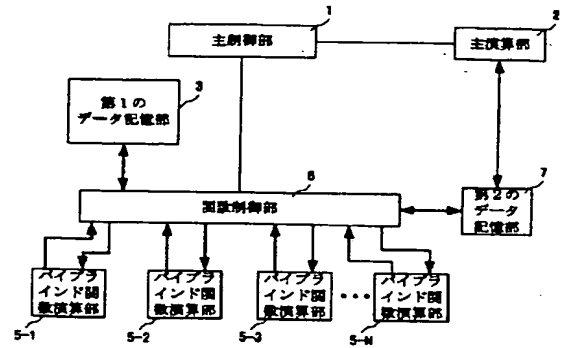
【符号の説明】

1…主制御部、2…主演算部、3…データ記憶部（第1のデータ記憶部）、5-1～5-N…バイブラインド関数演算部、6…関数制御部、7…第2のデータ記憶部、8…第3のデータ記憶部。

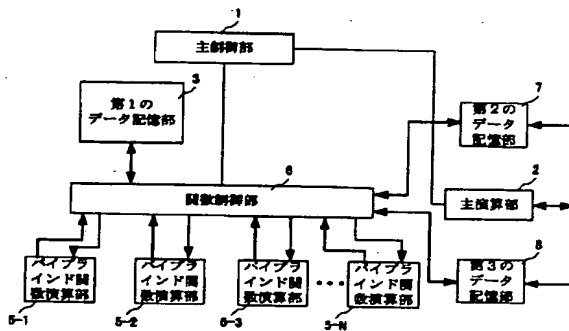
【図1】



【図2】



【図3】



【図5】

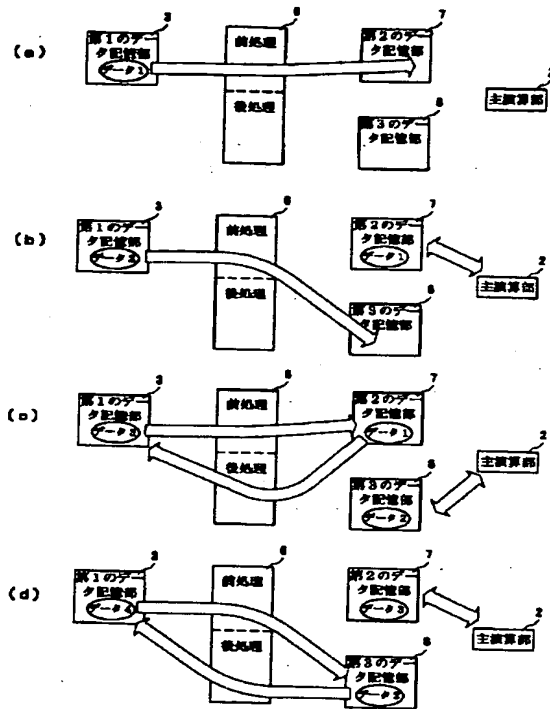
(a)

主演算部		FFT1		FFT2		FFT3	...	FFT128
第2のデータ記憶部	TR1	FFT1	TW1	TR2	FFT2	TW2	TR3	FFT3
								...
								FFT128
								TW128

(b)

主演算部		FFT1	FFT2	FFT3	...	FFT128
第2のデータ記憶部	TR1	FFT1	TW1	TR3	FFT3	...
第3のデータ記憶部		TR2	FFT2	TW2	TR4	...
						FFT128
						TW128

【図4】



【図6】

